

# AN ADAPTIVE, HIGH-ORDER PHASE-SPACE REMAPPING FOR THE TWO-DIMENSIONAL VLASOV-POISSON EQUATIONS

BEI WANG <sup>\*</sup>, GREG MILLER<sup>†</sup>, AND PHIL COLELLA<sup>‡</sup>

**Abstract.** The numerical solution of high dimensional Vlasov equation is usually performed by particle-in-cell (PIC) methods. However, due to the well-known numerical noise, it is challenging to use PIC methods to get a precise description of the distribution function in phase space. To control the numerical error, we introduce an adaptive phase-space remapping which regularizes the particle distribution by periodically reconstructing the distribution function on a hierarchy of phase-space grids with high-order interpolations. The positivity of the distribution function can be preserved by using a local redistribution technique. The method has been successfully applied to a set of classical plasma problems in one dimension. In this paper, we present the algorithm for the two dimensional Vlasov-Poisson equations. An efficient Poisson solver with infinite domain boundary conditions is used. The parallel scalability of the algorithm on massively parallel computers will be discussed.

**Key words.** Particle-in-cell (PIC) Methods, Adaptive Mesh Refinement, Phase-space Remapping, Numerical Noise, Vlasov-Poisson equation

**AMS subject classifications.**

**1. Introduction.** The Vlasov equation describes the dynamics of a species of charged particles under electromagnetic fields. In the electrostatic case, the normalized equation reads

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla_{\mathbf{x}} f + (-1)^s (\mathbf{E} + \mathbf{E}^e) \cdot \nabla_{\mathbf{v}} f = 0, \quad (1.1)$$

where  $f(\mathbf{x}, \mathbf{v}, t)$  is the distribution function of the species in phase space  $(\mathbf{x}, \mathbf{v}) \in \mathbb{R}^d \times \mathbb{R}^d$  with  $d = 1, 2, 3$ .  $s$  is 0 for positive charges and is 1 otherwise.  $\mathbf{E}$  and  $\mathbf{E}^e$  denote the self-consistent and the external electric field, respectively. This equation is the simplest model to study collisionless plasmas and beam propagation which is of importance to controlled thermonuclear fusion and accelerator modeling.

The Vlasov equation is a nonlinear hyperbolic equation in phase space so methods of solution can be guided by the well-established numerical analysis of classical partial differential equations. Accordingly, grid methods in fluid dynamics, such as transform methods, finite-volume methods, and semi-Lagrangian methods, can be employed. Operator splitting was successfully applied to the solution of the Vlasov equation by Cheng and Knorr [4] in 1970s. It reduces the solution of the multi-dimensional Vlasov equation to a set of one-dimensional advection problems, and therefore has become a widely used technique. Even with these well-established algorithms, performing high-dimensional simulations using grid methods is still a challenging task. The issue is the computational time and memory cost in dealing with the whole six dimensional phase space. With the advances of supercomputer, grids methods have achieved large development in the last decade. In semi-Lagrangian methods, Sonnendruker et al. [30] introduced the cubic spline method. Nakamura and Yabe [26] introduced the cubic interpolated propagation method. In finite-volume methods, Fijalkow [14]

<sup>\*</sup>Department of Applied Science, University of California, 1 Shields Avenue, Davis CA 95616. Present Address: Princeton Institute for Computational Science and Engineering, Princeton University, Princeton, NJ 08540. E-mail: beiwang@princeton.edu

<sup>†</sup>Department of Chemical Engineering and Materials Science, 1 Shields Avenue, Davis CA 95616.

<sup>‡</sup>Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, MS 50A-1148, 1 Cyclotron Road, Berkeley, CA 94720.

presented the flux balance method. A similar idea is used in a high-order finite-volume method based on mapped coordinate by Colella, Dorr and Hittinger [6]. Filbet, Sonnendrucker, and Bertrand [16] proposed the positive and flux conservative scheme using the idea of limiter.

A more widely used approach for the solution of the Vlasov equation is PIC methods [19, 1]. In PIC methods, the particles, a Lagrangian discretization of the distribution function, follow trajectories computed from the characteristic curves given by the Vlasov equation, whereas the self-consistent fields are calculated on a grid. Since the methods employ the fundamental equations without much approximation, it allows us to observe most of the physics in a plasma system with relatively few particles. However, as with all other particle methods, PIC methods suffer from numerical noise such that they have difficulty in simulating some problems, e.g., the problem with large dynamic ranges in velocity space. To remedy this deficiency, there are usually two approaches. One method is the so-called  $\delta f$  method [13, 28, 21], discretizing only the perturbation  $\delta f$  with respect to an equilibrium state  $f_0$  based on a particle method. The  $\delta f$  method has been successfully used in realistic applications, e.g., microtubulence in magnetic confined plasmas [22]. The limitation of this method is that it can only be applied to the problems which are close to equilibrium. An alternative approach is through periodically reconstructing the distribution function on a grid in phase space. Such remapping technique has been used in particle methods in fluid dynamics [8, 2], i.e., vortex methods and smoothed particle hydrodynamics (SPH), to maintain regularity of the particle distribution and thereby improve accuracy, but has much more limited use in PIC methods in plasma physics. It is worth mentioning that early work of Denavit [12] and more recent work of Vadlamani [31] and Yang [3] used the idea of remapping for PIC methods. However, They all used low order interpolation function which results in a first order method overall.

We studied a high order remapping scheme to PIC methods for the solution of the one-dimensional Vlasov-Poisson equations early [32]. Meanwhile, we provided a local redistribution technique such that the positivity of the distribution function could be preserved after high-order remapping. The initial numerical experiments on a set of classical plasma problems in one dimension are very encouraging. Remapping significantly reduces the numerical noise and results in a more consistent second-order convergence rate in the electric field error. We also investigated the effects of integrating mesh refinement to the uniform remapping. This is motivated by the observation that remapping, a numerical diffusive procedure, tends to create a large number of small-strength particles at the low density region of the distribution function. Mesh refinement has the potential to reduce this side effect.

In this paper, we extend the algorithm to the solution of the two-dimensional Vlasov-Poisson equations. This includes the use of an efficient Poisson solver with infinite domain boundary conditions for beam problems. High-dimensional simulations are very expensive with respect to memory usage. We perform the simulation on a parallel machine using domain decomposition in physical space. A scalable implementation based on domain decomposition in phase space will be discussed. We consider two types of numerical tests: plasma problems including linear Landau damping and the two stream instability, and a beam problem based on the paraxial model [15].

The rest of the paper is organized as follows. In §2, we first review the classical PIC methods for the Vlasov-Poisson equations. An efficient algorithm which solves the Poisson equation with infinite boundary conditions is described. Then we present the high-order and positive remapping on a hierarchy of locally-refined grids in two

dimensions. §3 discusses the parallel implementation of the algorithm. We show the numerical results in §4. Conclusion and future research will be given at the end.

## 2. Algorithms.

**2.1. PIC methods.** PIC methods are based on the Lagrangian description of the Vlasov equation

$$\frac{df(\mathbf{X}, \mathbf{V}, t)}{dt} = 0, \quad (2.1)$$

where the characteristics  $(\mathbf{X}(t), \mathbf{V}(t))$  are the solution of the equation of motion:

$$\frac{d\mathbf{X}}{dt} = \mathbf{V}(t), \quad \frac{d\mathbf{V}}{dt} = (-1)^s (\mathbf{E}(\mathbf{X}, t) + \mathbf{E}^e(\mathbf{X}, t)) \quad (2.2)$$

with initial conditions  $\mathbf{X}(t=0) = \mathbf{x}$  and  $\mathbf{V}(t=0) = \mathbf{v}$ .

In the beginning, the distribution function is approximated by a collection of point particles,

$$f(\mathbf{x}, \mathbf{v}, t=0) \approx \sum_k q_k \delta(\mathbf{x} - \mathbf{x}_k) \delta(\mathbf{v} - \mathbf{v}_k), \quad (2.3)$$

where  $(\mathbf{x}_k, \mathbf{v}_k)$  is a initial particle location at the cell center of a grid in phase space (quite start).  $q_k = f(\mathbf{x}_k, \mathbf{v}_k, t=0)h_x h_y h_{v_x} h_{v_y}$  is the weight of a particle. Then each particle follows a trajectory described by the equation of motion,

$$\frac{dq_k}{dt} = 0, \quad \frac{d\tilde{\mathbf{X}}_k}{dt}(t) = \tilde{\mathbf{V}}_k(t), \quad \frac{d\tilde{\mathbf{V}}_k}{dt}(t) = (-1)^s (\tilde{\mathbf{E}}_k(t) + \mathbf{E}_k^e(t)), \quad (2.4)$$

where  $\tilde{\mathbf{X}}_k(t=0) = \mathbf{x}_k$  and  $\tilde{\mathbf{V}}_k(t=0) = \mathbf{v}_k$ .

At any time that a smooth representation of the distribution function is required, we approximate the function with a collection of finite size particles, where the exact delta function is replaced by a smoothed delta function. That is,

$$f(\mathbf{x}, \mathbf{v}, t) \approx \sum_k q_k \delta_{\epsilon_x}(\mathbf{x} - \tilde{\mathbf{X}}_k(t)) \delta_{\epsilon_v}(\mathbf{v} - \tilde{\mathbf{V}}_k(t)), \quad t > 0. \quad (2.5)$$

The smoothed delta function satisfies

$$\int_{\mathbb{R}^2} \delta_{\epsilon}(\mathbf{y}) d\mathbf{y} = 1 \quad (2.6)$$

and

$$\delta_{\epsilon}(\mathbf{y}) = \prod_{d=0}^1 \frac{1}{\epsilon_d} u\left(\frac{y_d}{\epsilon_d}\right), \quad (2.7)$$

where  $u$  is any interpolation function and  $\epsilon$  is the stencil size. Usually, the stencil size for the smoothed delta function in physical space  $\epsilon_x$  is chosen as the same as the mesh spacing of the Poisson solver. The typical interpolation function for PIC methods is the first-order interpolation function

$$u_1(z) = \begin{cases} 1 - |z| & 0 \leq |z| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \quad (2.8)$$

The flow of a PIC scheme is

- Assign particle charges on a grid in physical space,

$$\tilde{\rho}(\mathbf{x}_j, t) = \sum_k q_k \delta_{\varepsilon_x}(\mathbf{x}_j - \tilde{\mathbf{X}}_k(t)), \quad (2.9)$$

where  $\mathbf{j} \in \mathbb{Z}^2$  are the node index of the Cartesian grid in physical space. The grid size is chosen as the same as the stencil size of the smoothed delta function  $\varepsilon_x$ . In the case of the first-order interpolation function, for each node  $\mathbf{j}$ , the sum is restricted to the particles with  $|\mathbf{x}_j - \tilde{\mathbf{X}}_k(t)| \leq \varepsilon_x$ .

- Solve the Poisson equation on the grid with a second-order finite-difference method:

$$-(\Delta^H \phi)_j = - \sum_{d=0}^1 \frac{\phi_{j+e^d} - 2\phi_j + \phi_{j-e^d}}{\varepsilon_{x_d}^2} = (-1)^s \tilde{\rho}_j + \rho_{\text{background}} \quad (2.10)$$

and

$$\tilde{\mathbf{E}}_j^d = \frac{\phi_{j-e^d} - \phi_{j+e^d}}{2\varepsilon_{x_d}}. \quad (2.11)$$

$\rho_{\text{background}}$  is the background charge density if applicable. With given boundary conditions, the discrete Poisson equation is usually solved by a fast Poisson solver, such as FFTs or multigrid methods.

- Interpolate the calculated field back to the particle locations with the same interpolation function in equation (2.7). It is worth mentioning that a different interpolation function will introduce self-force errors [7].
- Integrate the equation of motion numerically, for example, using the second-order Runge-Kutta method.

**2.2. Solving the Poisson equation with infinite domain boundary conditions.** To model beam problems, the Poisson equation with infinite domain boundary conditions needs to be solved. We compute the solution using a new version of the James-Lackner method [20] by McCorquodale et al. [23, 24]. This method solves two Dirichlet boundary problems plus a boundary to boundary convolution.

We briefly describe the algorithm below. Assume  $D_0$  is the support domain of the right-hand side  $\rho$ , we can solve the Poisson equation with infinite domain boundary conditions on a slightly larger domain  $D_1 > D$  with inhomogeneous Dirichlet boundary conditions. The boundary value can be calculated by Green's function convolution from the source  $\rho$  to the domain boundary  $\partial D_1$ . The volume source  $\rho$  to boundary  $\partial D_1$  convolution is relatively expensive, in particular for 3D problems. Instead of using a volume to boundary convolution, we can compute the boundary value by performing a boundary  $\partial D_1$  to boundary  $\partial D_2$  convolution and solving another Poisson equation on a domain  $D_2 > D_1 > D$  with Dirichlet boundary condition. The procedure of James' algorithm is (Figure (2.1)) :

- Step 1: Solve the Poisson equation on domain  $D_1$  with homogeneous Dirichlet boundary conditions

$$-\Delta \phi_1 = \rho \quad \text{on } D_1, \quad \phi_1 = 0 \quad \text{on } \partial D_1. \quad (2.12)$$

- Step 2: Calculate the surface charge on  $\partial D_1$

$$\partial \rho_1 = \frac{\partial \phi_1}{\partial \mathbf{n}} \quad \text{on } \partial D_1. \quad (2.13)$$

- Step 3: Perform a boundary to boundary convolution from  $\partial\rho_1$  to  $\partial D_2$

$$\partial\phi_2 = \int_{\partial D_1} G(\mathbf{x} - \mathbf{y}) \partial\rho_1(\mathbf{y}) dA_{\mathbf{y}} \quad (2.14)$$

using fast multiple methods [17].

- Step 4: Solve another Poisson equation on domain  $D_2$  with inhomogeneous Dirichlet boundary conditions

$$-\Delta\phi_2 = \rho \quad \text{on } D_2, \quad \phi_2 = \partial\phi_2 \quad \text{on } \partial D_2. \quad (2.15)$$

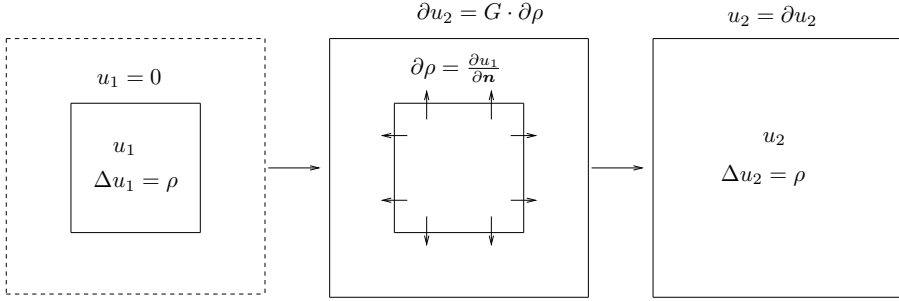


FIG. 2.1. *James Algorithm* Left: Solve for  $u_1$  on  $D_1$  Middle: Calculate surface charge and perform convolution Right: Solve  $u_2$  on  $D_2$

**2.3. Particle Remapping.** The convergence of particle methods for the one-dimensional Vlasov-Poisson equations has been investigated by Cottet and Raviart [9]. Their result shows that particle overlapping and regularization are important for the convergence of the methods. Specifically, the truncation error of a particle method is amplified by a time dependent exponential term. Based on Cottet and Raviart's work, we extend the error analysis to PIC methods [32]. Our result is one order higher in the truncation error. However, as in Cottet and Raviart's analysis, the truncation error is amplified by a time dependent exponential term. The analysis motivates the use of remapping technique, a widely used strategy in particle methods in fluid dynamics, to control the exponential error. The basic idea of remapping is simple. Since particles will gradually move away from the exact trajectories due to numerical error, we can reduce the displacement by periodically reproducing the distribution function  $f(\mathbf{x}, \mathbf{v}, t)$  on a grid by interpolation. A new set of particles, which are created from the grid representation, then replace the distorted particle distribution. The later step is identical to the initial step of PIC methods that we initialize the particle positions and weights in equation (2.3). The error due to remapping will depend on the order of the interpolation function.

In the previous work [32], we successfully applied the remapped PIC method to the one-dimensional Vlasov-Poisson system. The remapping scheme was extended in three aspects compared with the standard scheme. First, we used high-order interpolation functions which improve accuracy but do not preserve positivity. Second, we preserved the positivity of a high-order interpolation by redistributing the excess charge into its local neighborhood. The local redistribution algorithm is based on the mass redistribution idea of Chern and Colella [5], which is first applied to enforce positivity preservation by Hilditch and Colella [18]. Third, instead of reinitializing

on a uniform grid, we reproduced the distribution function on a hierarchy of locally-refined grids. Remapping on a hierarchy of locally-refined grids significantly reduces the number of small-strength particles located at the tail of the distribution function. The high-order, positive, and adaptive remapping scheme in high dimensional phase space is described below.

**2.3.1. High-order Remapping.** The overall accuracy introduced by remapping will be one order lower than the order of the interpolation function since we lose one order of accuracy in the evolution step. For example, the interpolation function with second-order accuracy only results in a first-order method overall. In this paper, we consider an interpolation function with third-order accuracy derived by Monaghan [25]. The function in one-dimensional can be expressed as

$$W_4(x, h) = \begin{cases} 1 - \frac{5s^2}{2} + \frac{3s^3}{2} & 0 \leq s = \frac{|x|}{h} \leq 1 \\ \frac{1}{2}(2-s)^2(1-s) & 1 \leq s = \frac{|x|}{h} \leq 2 \\ 0 & \text{otherwise.} \end{cases} \quad (2.16)$$

The one-dimensional expression can be generalized to four dimensions by tensor product,

$$W_4(\mathbf{x}_i - \mathbf{x}_k) = \prod_{d=0}^3 W_4(x_i^d - x_k^d, h^d), \quad (2.17)$$

where  $h^d$  is the remapping mesh spacing in phase space.  $i$  and  $k$  denote the index for the cell-centered grid and the particles, respectively.

This function, called a modified B-spline, conserves the total charge and represent a quadratic polynomial exactly. In addition, the first- and the second-order derivative of  $W_4(x, h)$  are continuous. The smoothness property of this modified B-spline is particularly good for scattered data interpolation. However, as with all other high-order interpolation functions,  $W_4$  is not positivity preserving. An interpolation function without positivity might create nonphysical negative charge. This should be avoided in simulations.

**2.3.2. Positivity.** The positivity preserving algorithm is based on the mass redistribution idea of Chern and Colella [5], first applied to enforce positivity preservation by Hilditch and Colella [18]. In the algorithm, we redistribute the undershoot of cell  $i$

$$\delta f_i = \min(0, f_i^n) \quad (2.18)$$

to its neighboring cells  $i + \ell$  in proportion to their capacity  $\xi$

$$\xi_{i+\ell} = \max(0, f_{i+\ell}^n). \quad (2.19)$$

The distribution function is conserved, which fixes the constant of proportionality

$$f_{i+\ell}^{n+1} = f_{i+\ell}^n + \frac{\xi_{i+\ell}}{\sum_{\mathbf{k} \neq 0} \xi_{i+\mathbf{k}}} \delta f_i \quad (2.20)$$

for  $\ell \neq 0$  such that cell  $i + \ell$  is a neighbor of cell  $i$ . Superscript  $n$  and  $n + 1$  denote the interpolated value before and after redistribution, respectively.

The drawback of this approach is that positivity is not guaranteed in a single pass. One might have to apply the method iteratively. In practice, however, we find a few iterations are sufficient.

**2.3.3. Mesh Refinement.** Mesh refinement is an attractive option in improving the efficiency of phase-space remapping. The distribution function in phase space is inhomogeneous, for example, a Maxwellian distribution in velocity space. When we represent the system by particles, it is best that we can have each particle carries a similar amount of weights. Remapping on a hierarchy of locally-refined grids is a good strategy for creating a such set of particles. From another point of view, remapping through interpolation is a numerically diffusive procedure. This results in a large number of small-strength particles near the tail of the distribution function. The situation becomes worse as we apply remapping frequently. Remapping on a hierarchy of locally-refined grids, with a coarser grid covering the tail of the distribution function, can reduce the number of those small-strength particles. In the following, we present the algorithm of remapping with mesh refinement. In designing the algorithm, we have two guiding principles: the total charge should be conserved and the overall accuracy on the field needs to be maintained.

Before explaining the algorithm, we introduce the definition of a composite grid. We define a hierarchy of cell-centered grids  $\Omega_\ell$ , where  $0 \leq \ell \leq \ell_{\max}$ .  $\Omega_{\ell=0}$  is the coarsest grid that covers the whole problem domain. The finer grids  $\Omega_{\ell>0}$  are constructed as a union of cell-centered rectangles (see Figure 2.2). The mesh spacing of each level is  $\mathbf{h}_\ell = \mathbf{h}_{\ell-1}/\mathbf{r}_{\ell-1}$ , where  $\mathbf{r}_{\ell-1}$  is the refinement ratio of level  $\ell-1$ . In four dimensions,  $\mathbf{h}_\ell \in \mathbb{R}^4$  and  $\mathbf{r}_\ell \in \mathbb{Z}^4$ . The composite grid consists of valid grids at all levels, where a valid grid is defined as a region not overlain by a finer grid. That is,

$$\Omega_c = \sum_{\ell=0}^{\ell_{\max}} (\Omega_\ell \setminus P_{\ell+1}^\ell(\Omega_{\ell+1})), \quad (2.21)$$

where  $P_{\ell+1}^\ell$  is the operator projecting from level  $\ell+1$  to level  $\ell$ .

At the beginning, a set of particles are created from the cell center of the composite grid. In the remapping step, each particle first finds the valid cell in the composite grid it belongs to. One particle can only belong to a single valid cell. If the cell is far enough away from a coarse-fine-interface, the charge can be interpolated on the grid as in equation (2.16). If the cell is near a coarse-fine interface such that the interpolation stencil intersects the coarse-fine interface, special care must be taken. First, we interpolate the charge on the surrounding cells as usual. After deposition, we know that not all deposited cells are valid cell. We need a further step to transfer the charge from invalid cell to valid cell. There are two cases depending on where the invalid cell is located. If the invalid cell is in a coarser level and it is covered by the valid cells of a finer level, we transfer the deposited charge from the coarser level to the finer level through interpolation. On another hand, if the invalid cell is outside the grid of the current level, the charge is transferred by projection. Figure 2.2 shows the algorithm in two dimensions. The four-dimensional case can be generalized easily. It is worth mentioning that we lose one order of accuracy in interpolating the coarser level charge into the finer level. However, since the coarse-fine interface is in co-dimension one, the expected accuracy in the field, e.g., second-order, will be preserved in  $L_\infty$  norm error. Our current implementation doesn't have time-dependent adaptivity. This feature can be incorporated by selecting some refinement criterion, for example each cell in phase space has similar number of particles.

**3. Parallel Implementation and Issues.** The parallel implementation of the algorithm is straightforward based on domain decomposition in physical space. The physical space is decomposed into  $M$  patches. Given a parallel machine with  $N$

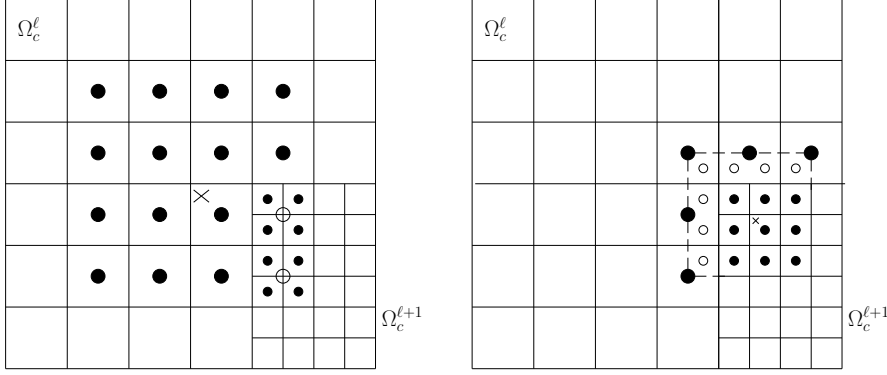


FIG. 2.2. Cross signs denote the particle locations. The valid and the invalid deposited cells are denoted by filled circles and open circles, respectively. The refinement ratio is  $\mathbf{r}_0 = (2, 2)$  in the plots. Left: Particle is at the coarser level side. Right: Particle is at the finer level side. Cross signs denote the particle locations.

processors, each patch is assigned to a processor cyclically. Particles are assigned to patches according to their physical space positions. Using MPI, patches communicate with each other through ghost-cells and particles move between patches. This is the default implementation in Chombo software [27].

The current implementation is not a scalable algorithm (weak scaling) because of decomposition in physical space only. The potential issue is that when the problem size increases, since the number of processors is scaled in proportion to the problem size in physical space, the computational time and memory usage will increase in proportion to the problem size in velocity space. In the worst case, the processor will be out of memory. An alternative implementation is based on domain decomposition in phase space. In this implementation, it might be the case that particles belong to the same cell in physical space (equation (2.9)) are distributed on different processors. Since the Laplacian operator is linear, we can choose to solve the Poisson equation separately on different processors. The total fields are then obtained by MPI\_Allreduce.

**4. Numerical Tests.** We demonstrate PIC methods with adaptive phase-space remapping on a set of classical plasma and beam problems in two dimensions, including linear Landau damping, the two stream instability, and beam propagation in the paraxial model. For satisfying the overlapping condition, we choose  $\varepsilon_x/h_x = \varepsilon_y/h_y = 2$ , where  $\varepsilon_{\mathbf{x}} = (\varepsilon_x, \varepsilon_y)$ .

We use Richardson extrapolation for error estimate. If  $\tilde{\mathbf{E}}^h$  is the electric field computed with the initial phase space discretization  $(h_x, h_y, h_{v_x}, h_{v_y})$  and integration step size  $\Delta t$ , and  $\tilde{\mathbf{E}}^{2h}$  computed with  $(2h_x, 2h_y, 2h_{v_x}, 2h_{v_y})$  and  $2\Delta t$ , the relative solution error in direction  $d$  is defined as

$$e_d^h = |\tilde{E}_d^h - \tilde{E}_d^{2h}|. \quad (4.1)$$

$q$  is the order of the method and is calculated by

$$q = \min_d \log_2 \left( \frac{\|\tilde{e}_d^{2h}\|}{\|\tilde{e}_d^h\|} \right). \quad (4.2)$$



**4.1. Linear Landau Damping.** The initial distribution for linear Landau damping is

$$f_0(x, y, v_x, v_y) = \frac{1}{2\pi} \exp(-(v_x^2 + v_y^2)/2)(1 + \alpha \cos(k_x x) \cos(k_y y)), \quad (4.3)$$

where  $\alpha = 0.05$ ,  $k_x = k_y = 0.5$  and  $v_{\max} = 6.0$ . The physical domain is  $(x, y) \in [0, L = 2\pi/k_x] \times [0, L = 2\pi/k_y]$  with periodic boundary conditions. Particle charges with strength less than  $1.0 \times 10^{-9}$  are ignored. In the simulation, we apply remapping every 5 PIC time steps.

In the first test, we are interested in the evolution of the amplitude of the electric field. According to Landau's theory, the electric field is expected to decrease exponentially with damping rate  $\gamma = -0.394$ . The behavior of exponential decay has been observed by many other authors, mostly calculated by grid methods [26, 16, 10, 11].

We initialize the problem on two levels of grids with base level at  $h_x = h_y = L/32$ ,  $h_{v_x} = h_{v_y} = v_{\max}/16$ . The velocity space is refined on sub-domain  $\mathbf{v} \in [-3, 3] \times [-3, 3]$  with a refinement ratio 2. The PIC step size is  $dt = 1/8$ . We compare the simulation with and without remapping in Figure (4.1). In the case with remapping, the computed damping rate is very close to the theoretical value. The simulation without remapping fails to track the exponential decay.

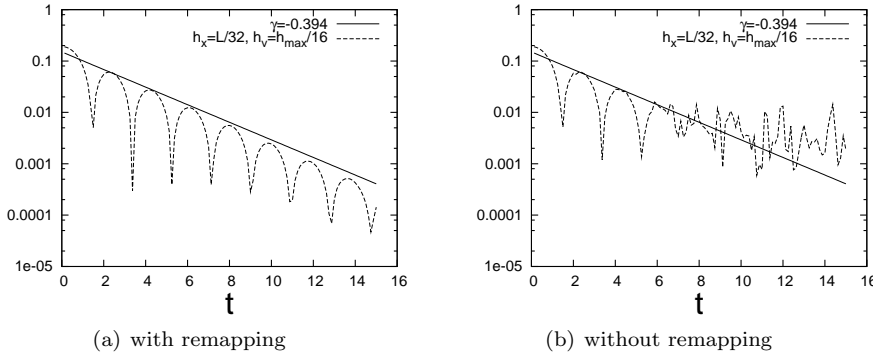


FIG. 4.1. The amplitude of the electric field for 2D linear Landau damping problem. Scales  $(h_x, h_{v_x})$  above denote the particle grid mesh spacing at the base level, where  $h_x = h_y$  and  $h_{v_x} = h_{v_y}$ . With remapping, the computed damping rate is very close to the theoretical value  $\gamma = -0.394$ . Without remapping, the simulation fails to track the exponential decay after a few damping circles.

In the second test, we compare the electric field errors and corresponding convergence rates with and without remapping in Figure (4.2) and (4.3). We see that remapping significantly reduces the electric field errors and improves their corresponding convergence rates.

**4.2. The Two Stream Instability.** The initial distribution for the two stream instability is

$$f_0(x, y, v_x, v_y) = \frac{1}{12\pi} \exp(-(v_x^2 + v_y^2)/2)(1 + \alpha \cos(k_x x))(1 + 5v_x^2), \quad (4.4)$$

where  $\alpha = 0.05$ ,  $k_x = 0.5$  and  $v_{\max} = 9.0$ . The physical domain is  $(x, y) \in [0, L = 2\pi/k_x] \times [0, L = 2\pi/k_y]$  with periodic boundary conditions. Particle charges with strength less than  $1.0 \times 10^{-9}$  are ignored. As in linear Landau damping problem, we

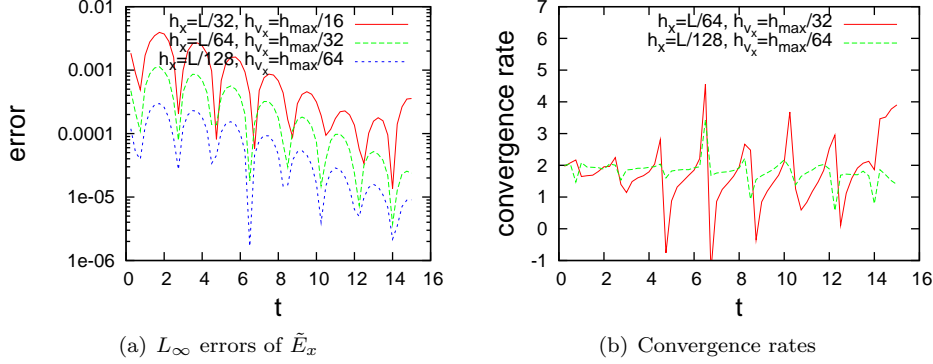


FIG. 4.2. Error and convergence rate plots for 2D linear Landau damping problem with remapping. Scales  $(h_x, h_{v_x})$  above denote the particle grid mesh spacing at the base level, where  $h_x = h_y$  and  $h_{v_x} = h_{v_y}$ . Second-order convergence rates are obtained. (a) the  $L_\infty$  norm of the electric field errors on three different resolutions. (b) the convergence rates for the errors on plot (a).

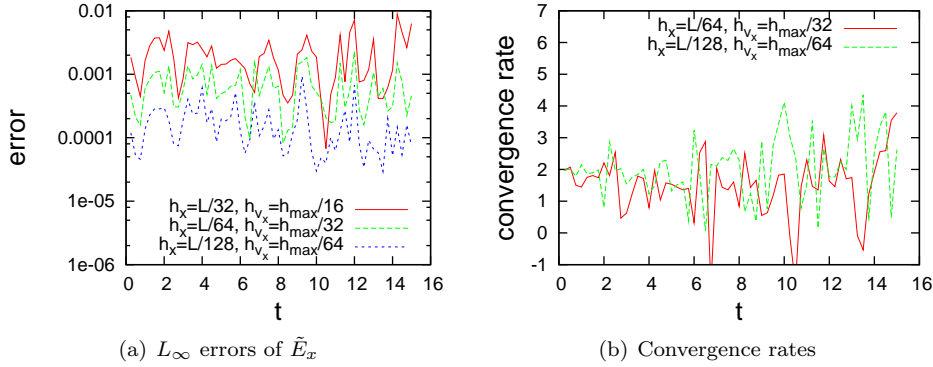


FIG. 4.3. Error and convergence rate plots for 2D linear Landau damping problem without remapping. Scales  $(h_x, h_v)$  above denote the particle grid mesh spacing at the base level, where  $h_x = h_y$  and  $h_{v_x} = h_{v_y}$ . The errors without remapping are much larger than the case with remapping (see Figure (4.2)). (a) the  $L_\infty$  norm of the electric field errors on three different resolutions. (b) the convergence rate for the errors on plot (a).

apply remapping every 5 PIC time steps. The velocity space is refined on sub-domain  $\mathbf{v} \in [-4.5, 4.5] \times [-4.5, 4.5]$  with a refinement ratio 2.

We compare the electric field errors and their convergence rates with and without remapping as before. Figure (4.5) shows the  $L_\infty$  norm of the errors at the case without remapping in three different resolutions. The corresponding convergence rates are shown on the right of the error plots. Second-order convergence rates are lost at the early time of the simulation. Comparing with the results with remapping in Figure (4.4), we see that remapping extends the second-order convergence rates to longer times.

We also compare the projected distribution function on plane  $(x, v_x)$  at the same instant time  $t = 20$  by both methods in Figure (4.6). For visualization purpose, in the case without remapping, we interpolate the particle-based distribution function to a

grid in phase space. We see that the classical PIC method results in a noisy solution (see Figure (4.6(b))). Figure (4.6(a)) shows the distribution function computed by the PIC method with remapping.

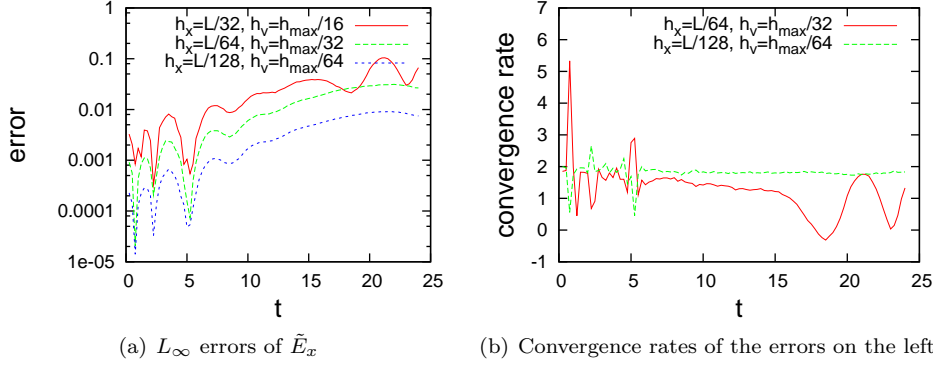


FIG. 4.4. Error and convergence rate plots for the two stream instability with remapping. Scales  $(h_x, h_v)$  above denote the particle grid mesh spacing at the base level, where  $h_x = h_y$  and  $h_{v_x} = h_{v_y}$ . The PIC step size is  $dt = 1/8$  at the lowest resolution. (a) the  $L_\infty$  norm of the electric field errors on three different resolutions. (b) the convergence rate for the errors on plot (a).

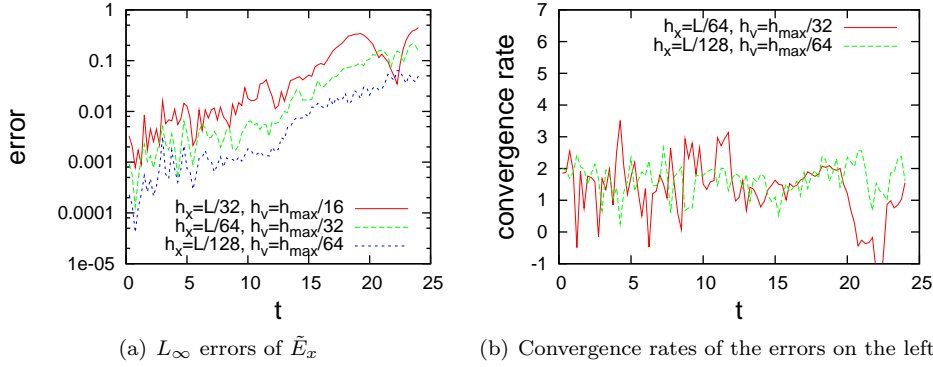


FIG. 4.5. Error and convergence rate plots for the two stream instability without remapping. Scales  $(h_x, h_v)$  above denote the particle grid mesh spacing at the base level, where  $h_x = h_y$  and  $h_{v_x} = h_{v_y}$ . The PIC step size is  $dt = 1/8$  at the lowest resolution. (a) the  $L_\infty$  norm of the electric field errors on three different resolutions. (b) the convergence rate for the errors on plot (a).

**4.3. Semi-Gaussian Beam.** The paraxial model is an approximation to the steady-state Vlasov-Maxwell equation in three dimensions. The K-V distribution is a measure solution of the paraxial model. Given an arbitrary initial distribution, we can focus a beam with the same matching forces for the K-V beam using the concept of equivalent beam. Here, we consider an initial semi-Gaussian beam focused by a uniform electric field using the concept of equivalent beam. The model has been considered by many authors [29, 10, 11].

In the test, the beam is composed of ionized potassium. The physical parameters are the following: current  $I = 0.2A$ , beam velocity  $v_b = 0.63 \times 10^6 m/s$ , and the radius

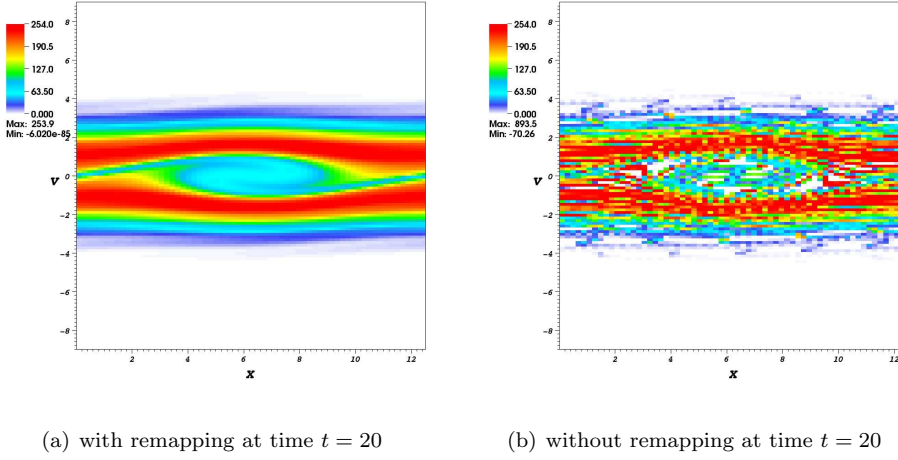


FIG. 4.6. Comparison of  $F(x, v_x)$ , the distribution function projected on space  $(x, v_x)$ , at the same instant of time  $t = 20$  with (Left) and without remapping (Right) for the two stream instability problem. The projected value is  $F(x, v_x) = \int_0^L \int_{-\infty}^{\infty} f(x, y, v_x, v_y) dy dv_y$ . The grid-based distribution function is obtained by reproducing the particle-based distribution function through a second-order interpolation. We initialize the distribution function on two levels of grids, with base level at  $h_x = h_y = L/64$ ,  $h_{v_x} = h_{v_y} = v_{\max}/32$ . The grid is refined by factor of 2 in velocity space on sub-domain  $\mathbf{v} \in [-4.5, 4.5] \times [-4.5, 4.5]$ . The classical PIC method results in a noisy solution with large errors in maximum. Both numerical noise and errors in maximum are significantly reduced by using remapping. The negative minimum in the case without remapping is a superficial effect due to project 4D data to 2D using a high-order interpolation for visualization purpose.

of the beam  $a = 0.02m$ . We choose the tune depression  $\eta = 1/2$ . For the normalization of the paraxial model, we refer to the work of Filbet and Sonnendrucker [15]. We use normalization parameters  $(x_0, v_0) = (a, \frac{\epsilon_x v_b}{2a})$ . This results in the normalized Poisson system

$$-\Delta\phi = \int_{\mathbb{R}^2} f dv, \quad -\nabla\phi = \mathbf{E}, \quad (4.5)$$

with initial semi-Gaussian distribution

$$f_0(x, y, v_x, v_y) = \begin{cases} \frac{4(1-\eta^2)}{\pi\eta^2} \exp(-(v_x^2 + v_y^2)/2), & x^2 + y^2 \leq 1 \\ 0, & \text{otherwise} \end{cases} \quad (4.6)$$

and the external matching field

$$E^e(x, y, t) = -\frac{4}{\eta^2}(x\mathbf{e}_x + y\mathbf{e}_y). \quad (4.7)$$

For the numerical parameters, we choose  $(h_x = |L_x|/128, h_y = |L_y|/128)$  and  $(h_{v_x} = v_{\max}/128, h_{v_y} = v_{\max}/128)$ , where  $(L_x, L_y) = (-10, 10)$  and  $v_{\max} = 10$ . The PIC time step is  $dt = 0.00052925$ .

Figure (4.7) shows the projection of the distribution function on planes  $(x, v_x)$  with and without remapping, respectively. The simulation with remapping gives a well-resolved result which preserves the positivity of the distribution function. Meanwhile, we show the root mean square (RMS) quantities of the semi-Gaussian beam

in Figure (4.8). Although the RMS quantities of the semi-Gaussian beam are oscillatory, they remain close to the quantities of the associated K-V beam and they converge as the resolution increases. As mentioned by the other authors [10], the oscillatory behavior is due to the fact that the semi-Gaussian is not exactly a steady state distribution.

**5. Conclusion.** In this paper, we have presented the adaptive remapped PIC method to the high-dimensional Vlasov equation and demonstrated in linear Landau damping, the two stream instability, and the beam propagation problems. The new method reduces the numerical noise significantly. There are two extensions of the current research. The first will be the development of a scalable algorithm based on domain decomposition in phase space. The second will be the introduction of time adaptivity to the current algorithm that the hierarchy of locally-refined grids are dynamically created from the particle distribution at every remapping step.

**Acknowledgments.** This work was supported by the U.S. Department of Energy Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231 at the Lawrence Berkeley National Laboratory. G. H. Miller was supported by DOE contract number de-sc0001981.

#### REFERENCES

- [1] C. K. BIRDSALL AND A. B. LANGDON, *Plasma physics via computer simulation*, Institute of Physics Publishing, Bristol and Philadelphia, 1991.
- [2] A. K. CHANIOTIS, D. POULIKAKOS, AND P. KOUMOUTSAKOS, *Remeshed smoothed particle hydrodynamics for the simulation of viscous and heat conducting flows*, J. Comput. Phys., 182 (2002), pp. 67–90.
- [3] YANG CHEN AND SCOTT E. PARKER, *Coarse-graining phase space in  $\Delta f$  particle-in-cell simulations*, Phys. Plasmas, 14 (2007), p. 082301.
- [4] C. Z. CHENG AND G. KNORR, *The integration of the Vlasov equation in configuration space*, J. Comput. Phys., 22 (1976), pp. 2330–2351.
- [5] I.-L. CHERN AND P. COLELLA, *A conservative front tracking method for hyperbolic conservation laws*, tech. report, Lawrence Livermore National Laboratory, 1987. UCRL-97200.
- [6] P. COLELLA, M. R. DORR, J. A. F. HITTINGER, AND D. F. MARTIN, *High-order finite-volume methods in mapped coordinates*, Submitted to J. Comput. Phys., (2010).
- [7] P. COLELLA AND P. C. NORGAARD, *Controlling self-force errors at refinement boundaries for AMR-PIC*, J. Comput. Phys., 229 (2010), pp. 947–957.
- [8] G.-H. COTTET AND P. D. KOUMOUTSAKOS, *Vortex Methods: Theory and Practice*, Cambridge University Press, Cambridge CB2 2RU, UK, 2000.
- [9] G.-H. COTTET AND P. A. RAVIART, *Particle methods for one-dimensional Vlasov-Poisson equations*, SIAM J. Numer. Anal., 21 (1984), pp. 52–76.
- [10] N. CROUSEILLES, M. GUTNIC, G. LATU, AND E. SONNENDRUCKER, *Comparison of two eulerian solvers for four-dimensional vlasov equation: Part ii*, Communications in nonlinear science and numerical simulation, 13 (2008), pp. 94–99.
- [11] N. CROUSEILLES, G. LATU, AND E. SONNENDRUCKER, *A parallel vlasov solver based on local cubic spline interpolation on patches*, J. Comput. Phys., 228 (2009), pp. 1429–1446.
- [12] J. DENAVIT, *Numerical simulation of plasma with periodic smoothing in phase space*, J. Comput. Phys., 9 (1972), pp. 75–98.
- [13] A. M. DIMITS AND W. W. LEE, *Partially linearized algorithms in gyrokinetic particle simulation*, J. Comput. Phys., 107 (1993), pp. 309–323.
- [14] E. FIJALKOW, *A numerical solution to the Vlasov equation*, Comput. Phys. Comm., 116 (319–328), p. 1999.
- [15] F. FILBET AND E. SONNENDRUCKER, *Modeling and numerical simulation of space charged dominated beams in the paraxial approximation*, Math Mod. Meth. Appl. Sci., 16 (2006), pp. 763–791.
- [16] F. FILBET, E. SONNENDRUCKER, AND P. BERTRAND, *Conservative numerical schemes for the Vlasov equation*, J. Comput. Phys., 172 (2001), pp. 166–187.

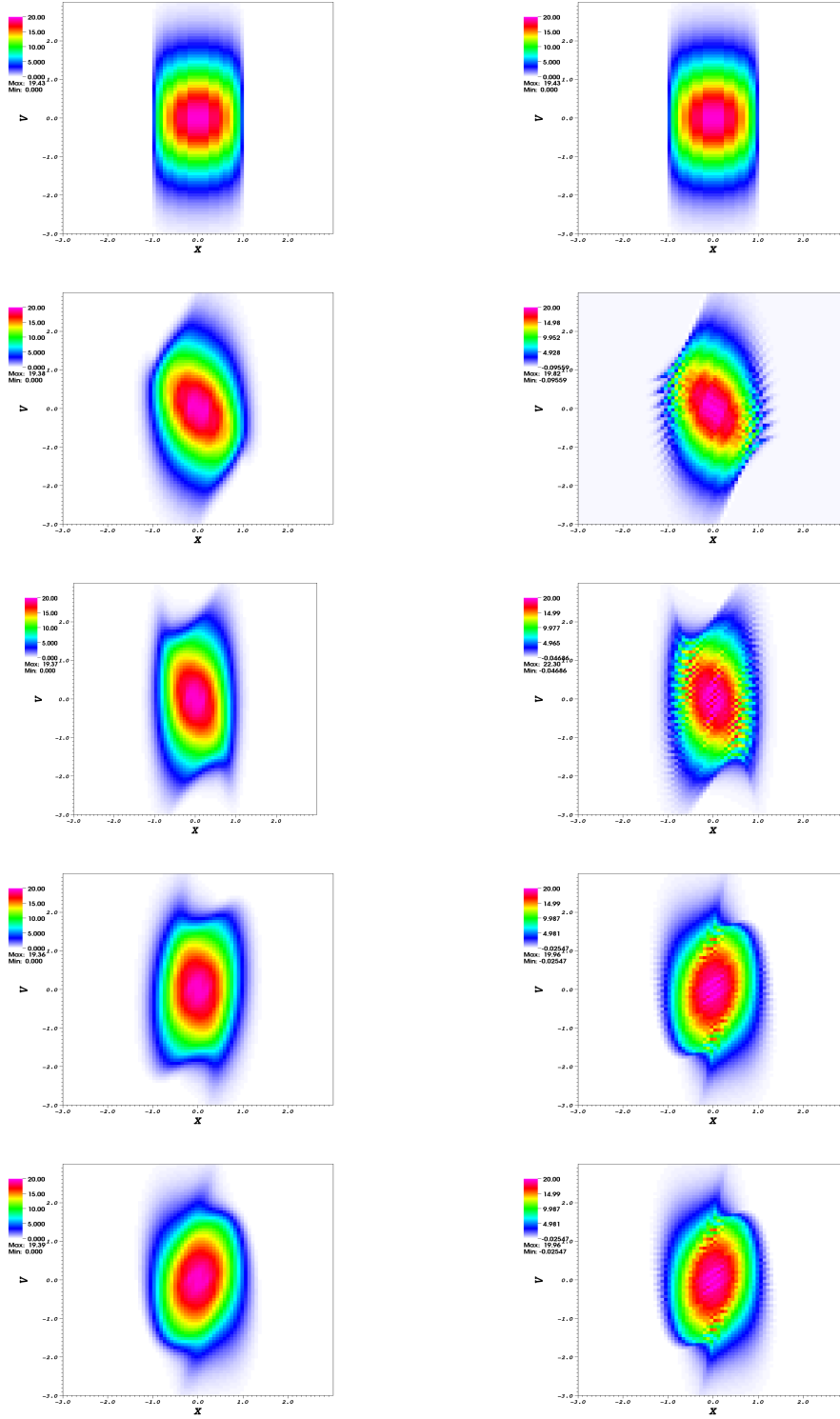


FIG. 4.7. Time evolution of the projection of the distribution function on  $(x, v_x)$  at time  $t = 0, 0.3176, 0.5239, 0.7410, 0.9527$  (from top to bottom). The projected value is  $F(x, v_x) = \int_0^L \int_{-\infty}^{\infty} f(x, y, v_x, v_y) dy dv_y$ . The grid-based distribution function is obtained by reproducing the particle-based distribution function through a second-order interpolation. The columns on the left and on the right are simulations running with and without remapping, respectively. The classical PIC method results in a noisy solution with large errors in maximum.

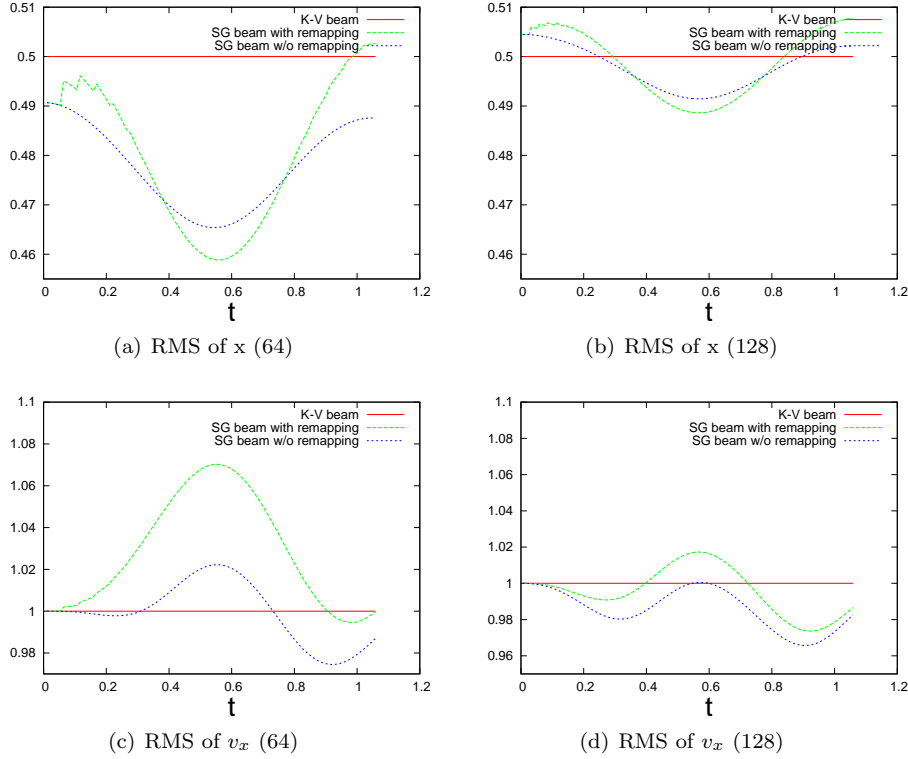


FIG. 4.8. Comparison of RMS quantities of semi-Gaussian beam with remapping, without remapping, and its equivalent K-V beam. We show the quantities in two different resolutions (64 vs 128 grid size in each dimension). Although the values are oscillatory, they remain close to the equivalent K-V beam and they converge to the equivalent K-V beam as the resolution increases.

- [17] LESLIE GREENGARD, *The rapid evaluation of potential fields in particle systems*, PhD thesis, Yale University, 1987.
- [18] J. HILDITCH AND P. COLELLA, *A projection method for low Mach number fast chemistry reacting flow*, in Proc. AIAA Aerospace Sciences Meeting, Reno, NV., 1997.
- [19] R. W. HOCKNEY AND J. W. EASTWOOD, *Computer Simulation Using Particles*, McGraw-Hill, New York, 1981.
- [20] R. A. JAMES, *The solution of Poisson's equation for isolated source distributions*, J. Comput. Phys., 25 (1977), pp. 71–93.
- [21] W. W. LEE, T. G. JENKINS, AND S. ETHIER, *A generalized weight-based particle-in-cell simulation scheme*, Comput. Phys. Comm., 182, pp. 564–569.
- [22] Z. LIN, T. S. HAHM, W. W. LEE, M. TANG W, AND R. B. WHITE, *Turbulent transport reduction by zonal flows: massively parallel simulations*, Science, 281, pp. 1835–1837.
- [23] P. MCCORQUODALE, P. COLELLA, G. T. BALLS, AND S. B. BADEN, *A scalable parallel Poisson solver in three dimensions with infinite-domain boundary conditions*, In 7<sup>th</sup> International Workshop on High Performance Scientific and Engineering Computing, (2005), pp. 814–822.
- [24] —, *A local corrections algorithm for solving Poisson's equation in three dimensions*, Comm. App. Math. and Comp. Sci., 2 (2007), pp. 57–81.
- [25] J. J. MONAGHAN, *Particle methods for hydrodynamics*, Comput. Phys. Rep., 3 (1985), pp. 71–124.
- [26] T. NAKAMURA AND T. YABE, *Cubic interpolated propagation scheme for solving the hyper-dimensional Vlasov-Poisson equation in phase space*, Comput. Phys. Comm., 120 (1999), pp. 122–154.
- [27] APPLIED NUMERICAL ALGORITHMS GROUP OF LAWRENCE BERKELEY NATIONAL LABORATORY,

- Chombo software.* <https://commons.lbl.gov/display/chombo/>.
- [28] S. PARKER, W. LEE, AND R. SANTORO, *Gyrokinetic simulation of itg driven turbulence in 3d toroidal geometry*, Phys. Rev. Lett., 71 (1993), pp. 2042–2045.
  - [29] E. SONNENDRUCKER, F. FILBET, A. FRIEDMAN, E. OUDET, AND J.-L. VAY, *Vlasov simulations of beams with a moving grid*, Comput. Phys. Comm., 164 (2004), pp. 390–395.
  - [30] E. SONNENDRUCKER, J. ROCHE, P. BERTRAND, AND A. GHIZZO, *The semi-Lagrangian method for the numerical resolution of Vlasov equations*, J. Comput. Phys., 149 (1998), pp. 201–220.
  - [31] SRINATH VADLAMANI, SCOTT E. PARKER, YANG CHENG, AND CHARLSON KIM, *The particle-continuum method: an algorithmic unification of particle-in-cell and continuum methods*, Comput. Phys. Comm., 164 (2004), pp. 209–213.
  - [32] B. WANG, G. MILLER, AND P. COLELLA, *A particle-in-cell method with adaptive phase-space remapping for kinetic plasmas*, SIAM in Sci. Comput., 33 (2011), pp. 3509–3537.